Project descriptions

There are two types of projects: Type-A and Type-B. Type-A is an individual project, with two pre-designed projects: A1. AlexNet and A2. LSTM. Project A1 is only for undergrad students and A2 is for graduate students but interested undergrad student can also select it. For Type-B, you can name your own projects. The detailed descriptions for these projects are shown below.

Type-A projects

A1. Understanding and Implementing AlexNet for Image Classification

Objectives: In this project, students will implement and modify AlexNet to classify colour images using the CIFAR-10 dataset. You will explore deep learning concepts such as convolutional layers, dropout, optimization techniques, and visualization.

Requirements:
1) Read the original AlexNet paper: "ImageNet Classification with Deep Convolutional Neural Networks" (Krizhevsky et al., 2012); Compare CIFAR-10 with ImageNet (AlexNet's original dataset) and discuss the differences as you need to modify AlexNet for the CIFAR-10 dataset.

2) Use PyTorch, TensorFlow or other DL frameworks you find appropriate to implement and modify AlexNet; **no need to implement it manually**.

3) Hyperparameter tuning & optimization: experiment with different learning rates (e.g., 0.01, 0.001, 0.0001), batch sizes (e.g., 32, 64, 128), optimizers (SGD, Adam, RMSprop), with dropout and without dropout, with batch normalization (BN) and without BN.

4) Record, compare, and discuss the corresponding results. Try to improve the classification accuracy, which will be positively correlated to your final grade.

Deliverables: Submit a **PDF** report summarizing model interpretations, implementation, experiments, results, and discussions. Source codes need to be attached **in the end of this report**.

A2. Understanding and Implementing LSTM from Scratch

Objectives: The goal of this project is to gain a deep understanding of Long Short-Term Memory (LSTM) networks by implementing LSTM manually. "Manually" means that you cannot use the built-in LSTM libraries like torch.nn.LSTM in Pytorch, tf.keras.layers.LSTM in tensorflow, LSTM in MatLab Deep Learning Toolbox, ect., to call LSTM directly; you need to construct a LSTM network from scratch.

Requirements:
1) Read the original LSTM paper "Long Short-Term Memory" by Hochreiter & Schmidhuber (1997) or other papers that introduce LSTM; understand LSTM, especially the gating mechanism and training mechanism in an LSTM.

2) Manually implementing LSTM from scratch: use only basic matrix operations and activation functions; implements a single LSTM cell that processes one timestep at a time, and then extending LSTM Cell to a full sequence.

3) Train your LSTM on the following task using the attached dataset "Car_data.zip":

- Dataset Description: The dataset "Car data.zip" consists of 18,457 files. Each file contains data for 12 labels, observed over 67 time steps. The data include: i) Coordinates (x, y) of the car; ii) Velocity; iii) Acceleration; iv) Distance from the centerline; v) Indicators if there's a car on the left, right, or front; vi) Distance from your vehicle to other cars.
- Prediction Task: Your LSTM model should be trained to predict the (x, y) coordinates of the vehicle for the upcoming 5 time steps.
- Error Rate: Evaluate the performance of your model using suitable error metrics like the Root Mean Square Error (RMSE) or any other loss function you find appropriate.
- Data Usage: While the dataset provides a range of data labels, you can decide how many labels you want to use. You might choose to use only the car's coordinates, or you could incorporate more information to potentially enhance your model's predictions. Also, determine the portion of training, validation, testing dataset by yourself.
- Data Continuity: Although the data is continuous across many files, it does restart at random points. It is recommended to consider each file (12 labels

x 67 observations) as an individual training unit. For example, use the first 12 labels x 62 observations as the input and the corresponding 2 labels x 5 observations as the output to train the LSTM network.

Note that it is very flexible to process the data and the way of processing them depends on you.

4) Record the prediction performance using the testing dataset you divide in the step 3).

5) Adjust the hyperparameters and train again, and then record, compare, and discuss the corresponding results. Try to improve the prediction accuracy, which will be positively correlated to your final grade.

Deliverables: Submit a **PDF** report summarizing model interpretations, implementation, experiments, results, and discussions. Source codes need to be attached **in the end of this report**.

Type-B projects

- You can either work by yourself or in a group of 2.
- You can propose a topic for your Type-B project. The topic of Type-B project should be about some advanced topics of neural networks and its applications, but its complexity should be similar (for individual) or higher (for group of 2) than Type-A project.
- A one-page project proposal is needed if you would like to name your own Type-B project. The project proposal should include: the motivation of this topic, the used data, the state-of-the-art approaches and your adopted model, evaluation metrics.
- The project proposal should be submitted and approved by instructor by **April 13**.
- The final submission for this project includes a **PDF** report that interprets your selected model and describes how you implement it. Also, include experiments, results, and discussions in the report. The results for different hyperparameters should be exhibited. Moreover, source codes need to be attached **in the end of this report**.